Drawings

Applicant submits herewith formal drawings for entry in the case subject to the approval of the Examiner.

35 USC 103

Claims 1 and 31-39 have been rejected under 35 U.S.C. 103(a) over Gupta et al. (US Patent 5,535,116, hereafter Gupta) in view of Hagersten et al. (US Patent 5,887,138, hereafter Hagersten).

Applicant traverses.

Each of applicant's claims recite (1) a shared memory where sections of the same global memory are located at different nodes, (2) a system of cache coherency over a multi-stage network (not a bus), and (3) a cache coherency where shared memory always contains the latest copy of changed data.

With respect to Gupta, as the Examiner states in the Office Action, "Gupta fails to specifically teach the

features... each processing node including a unique section of shared memory which is not a cache and an adapter for storing changed data to each unique section so that each section contains the most recent data." However, the Examiner states that Hagersten teaches these features.

The Examiner correctly states that Hagersten teaches a plurality of processing nodes interconnected through interconnecting network; each processing node including multiple processors, multiple cache memories and local memory, all local memories or memory portions of the processing nodes collectively form a distributive shared memory which may be accessed in non-uniform memory architecture (NUMA) fashion, i.e. each of said multi-processing nodes includes an addressable portion or local memory modules of the global system memory or sub-divided portion of the global physical system memory, and each processing node having said local memory is capable of storing valid and shared copies of requested ones of data signals stored in the main memory modules. The Examiner further correctly states that Hagersten in many instances indicates that each node of the plurality of nodes shares a shared distributed memory wherein sections or addressable locations of the shared distributed memory are accessible to more than one of the processing nodes. Hagersten's system

does not simply provides bus operations, but provides a coherency system maintaining coherency between the memories within the multi-processing nodes communicating via an interconnecting network.

Further, the Examiner is correct regarding Hagersten having "external device owning exclusive copy can unilaterally modify the copy without having to inform other entities". However, this is for an inter-node or processor-bus transaction within a subnode since all processors within a node have respective cache memory, but common local memory or system memory portion. Hagersten's system maintains internode coherency detecting addresses which require data transfer to or from another processing node; i.e., accessed addresses within the address space of a node corresponding to addresses within locations of another processing are maintained in such a way as to maintain consistency such that modified copies of data are written through so that all nodes maintaining a shared copy are allowed to cache the most current copy of a modified memory block.

However, in spite of the common features shared by the present invention and Hagersten, Applicant asserts that the present application still teaches and claims significant distinguishing features as follows.

1) Cache Coherency Method Differences

Hagersten teaches a write-back coherency operation which floods the network with excessive traffic, whereas the present application teaches a write-thru cache, which provides a better solution and requires much less network traffic.

For the present invention, the write-thru feature is claimed in every independent claim with statements similar to: "changing said shared memory, wherein changed data is stored immediately to shared memory regardless of which of said nodes is changing the data and which of said nodes includes the section of shared memory to be changed, wherein said shared memory always contains the most recent data." On the other hand, Hagersten et al. teaches write-back coherency, where a "requesting node" must query a "home node" for the location of which node and cache contains the most recent data. The "home node" searches for the location of the cache holding the requested data amongst "slave nodes" (i.e., all nodes having a cached copy of the requested data). Then, all slaves nodes must respond to the requesting node, where the "owning" slave node sends the latest value of the requested data and the other slave nodes

acknowledge that they also have a copy of the data. Hagersten calls this coherency process a "three-hop communication" [col.5, line 42]. These hops are as follows: hop 1) requesting node to home node, hop 2) home node to all slave nodes having a copy of the data in search of the most recent copy, and hop 3) all slaves nodes to the requesting node. Hagersten explains this three-hop process in col.5. lines 29-43.

The present invention does not teach a three-hop or four-hop method (four hop is referenced by Hagersten at col. 4, line 45), where the cache of any slave node can store the most recent data. In contrast, the present application teaches a <u>write-thru</u> cache coherency method, whereby any node changing the data immediately writes the most recent data back to the shared memory at the home node, such that the shared memory at the home node always has the most recent data, and the home node never has to search any other nodes or caches for where the most recent data is stored. In contrast with Hagersten, the present invention implements a two-hop method as follows: hop 1) a requesting node requests the most recent data of the home node, and hop 2) the home immediately returns the most recent data from its shared memory to the requesting node. This provides the quickest response possible for fetching global data - there

can be nothing faster or simpler than the method taught by the present application. Therefore, it is a unique and vast improvement over Hagersten.

2) Network Differences

Hagersten teaches point-to-point network 14 as shown in Figure 1, where each node 12A-D communicates over a direct link to each other node 12A-D over network 14. Hagersten network 14 is designed to allow his cache coherency method to work more efficiently. Suppose, for example, Hagersten node 12A is the requesting node that requests data, node 12B is the home node, and nodes 12C-D are the slaves nodes. For hop 1, a network transaction takes place between 12A to 12B on network 14. For hop 2, a network transaction takes place between 12B to 12C and 12D on network 14; however, since 12B has parallel point-to-point connections through network 14 to 12C and 12D, the communications can take place simultaneously while leaving the rest of the network 14 connections open for other network traffic. For hop 3, network transactions take place between 12C to 12A and 12D to 12A on network 14; however, since 12A has parallel point-to-point connections through network 14 to 12C and 12D, the

communications can take place simultaneously while leaving the rest of the network 14 connections open for other network traffic. Thus, Hagersten's cache coherency method works faster on his network since the network is designed for parallel communications over direct links between nodes.

However, the drawback of Hagersten's network is that it is not a good network for expanding to many nodes, because as the number of nodes n increases, the number of point-to-point connecting links (and therefore cables) increases by $n[(n-1)!]$ and the complexity of the System Interface block 24 of Figure 1 increases by n. This means Hagersten's method becomes very unwieldy with an increase in nodes. In contrast, the present invention is designed for a different type of network, a multi-stage network shown in Figure 4 of the present invention, where each node has one and only one connection to the network and the network is designed for expandability to hundreds or thousands of nodes. Note that each claim of the present clearly states that the present invention is for a multi-stage network, not a point-to-point network. Thus, Hagersten's solution is a limited solution that applies well to a small number of nodes and works more efficiently on a parallel point-to-point network.

Hagersten's method can run on a multi-stage network but all communications of home nodes to slave nodes (hop 2) and slave nodes to requesting nodes (hop 3) would have to take place in a serial fashion as each node has one and only one serial connection to a multi-stage network. The serial communication aspect of transfers from or to any given node of a multi-stage network plus the larger number of communications required in searching for the slave nodes and having all slave nodes respond, would cause Hagersten's method to perform very poorly on a multi-stage network.

The claims of the present invention require no further modification as they already clearly differentiate the unique aspects over Hagersten of write-thru caches storing the most recent data directly to shared memory and communication over a multi-stage network.

In conclusion, neither Hagersten nor Gupta, nor their combination, teaches applicant's invention as claimed.

SUMMARY AND CONCLUSION

Applicant urges that the case be passed to issue with claims 1 and 31-39.

If, in the opinion of the Examiner, a telephone conversation with applicant's attorney could possibly facilitate prosecution of the case, he may be reached at the number noted below.

Sincerely,

Howard T. Olnowich

By

Shelley M Beckstrand

Shelley M Beckstrand
Reg. No. 24,886

Date: 21 Jan 2002

Shelley M Beckstrand, P.C.
Attorney at Law
314 Main Street
Owego, NY 13827

Phone: (607) 687-9913
Fax: (607) 687-7848